



TITLE:

完全マッチング数え上げの高速な 指数時間アルゴリズムについて (ア ルゴリズムと計算理論の新展開)

AUTHOR(S):

泉, 泰介; 和田山, 正

CITATION:

泉, 泰介 ...[et al]. 完全マッチング数え上げの高速な指数時間アルゴリズムについて (アルゴリズムと計算理論の新展開). 数理解析研究所講究録 2012, 1799: 1-12

ISSUE DATE:

2012-06

URL:

<http://hdl.handle.net/2433/173010>

RIGHT:

2011 年度冬の LA シンポジウム [1]

完全マッチング数え上げの 高速な指数時間アルゴリズムについて

泉 泰介*

和田山 正†

概要

与えられたグラフにおける完全マッチングの個数の計数することは数え上げ問題における代表的な困難問題であり、グラフを 3-正則二部グラフに制限したもとでも #P-完全であることが知られている。本研究は、3-正則二部グラフの完全マッチング計数問題に対する、 $\tilde{O}(2^{5n/12})$ 時間の多項式メモリスペースアルゴリズムを提案する。このアルゴリズムは、グラフの閉路空間およびカット空間に対する符号理論的な解釈に基づいており、線形符号における主符号と双対符号の間の関係式（マックウィリアムス恒等式）を利用して完全マッチング計数問題をカットの重み分布計算に帰着することで高速なアルゴリズムを実現している。

1 はじめに

入力として与えられたグラフ G に対して、その完全マッチングの個数を計数する問題は数え上げ問題において重要な意味を持つ基本問題である。特に、 G が二部グラフの場合における完全マッチングの計数は、 $n \times n$ 行列の 0-1 パーマネントの計算と等価であることが知られている。パーマネント計算は統計物理学等で重要な応用を持つが、その計算は #P-完全と呼ばれる困難問題のクラスに属しており [24], n の多項式時間で値を計算することはほぼ絶望視されている。また、この困難性はグラフを 3-正則二部グラフに制限したもとでも成立することも知られてい

る [7]。上記の困難性のため、完全マッチング計数問題は主に 1) 近似アルゴリズムの設計、2) (高速な) 指数時間アルゴリズムの設計、の 2 アプローチから研究されている。本研究は、特に同問題に対する指数時間アルゴリズムの設計に着目する。

二部グラフに対する完全マッチング計数（すなわち、パーマネントの計算）のための指数時間アルゴリズムとして最もよく知られているものは、Ryser によるアルゴリズムである [21]。このアルゴリズムは、包除原理を利用することで、完全マッチングの個数を項数 $2^{n/2}$ の閉じた多項式で表現できるという事実を利用しており、その計算時間は $\tilde{O}(2^{n/2})$ 、メモリ使用量は $\text{poly}(n)$ ビットである¹。Ryser のアルゴリズム以降、計算時間が $\tilde{O}(2^{n/2})$ を下回る二部グラフの完全マッチング計数アルゴリズムについては長い間考えられていなかったが、2002 年に Bax と Franklin による期待計算時間 $\tilde{O}(2^{n/2 - o(n)})$ 、メモリ使用量 $\text{poly}(n)$ ビットの乱択アルゴリズムが示された [2]。このアルゴリズムは Servedio と Wan により計算時間 $\tilde{O}((2 - \epsilon)^{n/2})$ の決定性アルゴリズムに改良されている [22]。ここで ϵ はグラフの辺密度 m/n により決まる値である。特にグラフが 3-正則の場合、 $\epsilon \approx 0.002$ となり、Servedio と Wan のアルゴリズムは時間計算量 $\tilde{O}(1.4135^n)$ となる。多項式メモリスペースのアルゴリズムとしては、この結果が 3-正則二部グラフに対する現時点での最速アルゴリズムである。指数メモリスペースを利用することを許した場合、3-正則グラフのパス幅の上界が $n/6$ であるという事実 [9] と、木幅に対する固定パラメータ容易

*名古屋工業大学大学院情報科学研究科
†第 1 著者に同じ

¹ $\tilde{O}(f(n))$ は、通常のオーダ記法から $f(n)$ に対する対数オーダの因子（すなわち $\text{polylog}(f(n))$ ）を省略した記法である。

アルゴリズム [1] と組み合わせることで、 $\tilde{O}(2^{n/6})$ 時間のアルゴリズムを得ることができる。

本研究では、グラフの完全マッチング計数に対する新たなアプローチを提案する。提案する手法の基盤となるアイデアは、入力として与えられたグラフに対応する閉路空間およびカット空間を線形符号と見なすことで、符号理論における手法を利用して完全マッチング計数問題をカット空間の重み分布計算に帰着することである。より具体的には、提案手法は以下の3つの事実から構成されている。

- 任意のグラフ G に対して、定数個の頂点および高々 $n+1$ 本の辺を追加することで、 G と完全マッチングの個数が等しい奇グラフ \hat{G} を構成できる。また、任意の二部グラフ G に対して、定数個の頂点および高々 $n+5$ 本の辺を追加することで、 G と完全マッチングの個数が等しい奇グラフ \hat{G} を構成できる。
- 頂点数 n 、辺数 m の奇グラフにおいては、完全マッチングの数が、閉路空間における重み $m-n/2$ の符号語 (すなわち、辺数 $m-n/2$ の閉路空間中の要素) の数に等しい。
- 閉路空間およびカット空間の間に主双対関係があり、符号理論における主符号および双対符号の間の重み分布に関して成立する関係式 (マックウィリアムスの恒等式) を利用することで、閉路空間における重み $m-n/2$ の符号語数の計算をカット空間の重み分布計算に帰着することが可能である。

上記は一般のグラフにたいして成立する事実である。本研究では、それに併せて、3-正則二部グラフに対するカット空間の重み分布を、多項式メモリスペース $\tilde{O}(2^{5n/12})$ 時間で計算するアルゴリズムを提案する。このことは同時間、同メモリ使用量で3-正則二部グラフに対する完全マッチングの数え上げが可能であることを意味する。

本論文の構成は以下の通りである。まず、次節にて関連研究を述べたのち、3節にて線形符号に関する基本的な事項を述べる。4節および5節にて完全

マッチング計数問題と線形符号の関係について述べる。この節において完全マッチング計数問題のカット重み分布への帰着を与える。6節ではカット重み分布の計算を行う $\tilde{O}(2^{5n/12})$ 時間アルゴリズムを与える。最後に7節において本論文の結果のまとめを述べる。

2 関連研究

前節で述べた通り、完全マッチングの数え上げについては、近似アルゴリズムの設計、および (指数時間) 厳密アルゴリズムの両面から研究されてきた。二部グラフに限定した場合の結果については既に述べているので、本節では、一般のグラフおよび二部グラフ以外のグラフクラスに対する完全マッチング計数問題、およびその計算困難性についての話題を述べる。

二部グラフの完全マッチング計数がパーマネント計算との関連から比較的早くより研究されてきたことに対し、一般のグラフに対する完全マッチングの計数については自明なアルゴリズム (頂点集合の均等二分分割を全列挙するアルゴリズム) より真に高速なアルゴリズムについてはあまり検討されていなかった。しかし、近年、Björklund と Husfeldt により [5], $\tilde{O}(2^n)$ 時間の多項式メモリスペースアルゴリズム、および $\tilde{O}(1.733^n)$ 時間の指数メモリスペースアルゴリズムが示されて以降、急速にその計算量が改善されている。Koivisto は指数メモリスペースアルゴリズムにおける上記の計算量を $\tilde{O}(1.618^n)$ 時間に改善し [17], Nederlof は多項式メモリスペースアルゴリズムの上界を $\tilde{O}(1.942^n)$ に改善している [18]。Nederlof はまた、サイズ i の独立点集合を持つグラフについて、 $\tilde{O}(2^{n-i})$ 時間で動作するアルゴリズムを示している [19]。ごく最近、Björklund により、上記のアルゴリズムすべての性能を上回る、 $\tilde{O}(1.414^n)$ ($= \tilde{O}(2^{n/2})$) 時間、多項式メモリスペースのアルゴリズムが示された [4]。

制限されたグラフクラスに対する多項式時間厳密アルゴリズムの構成についてもいくつかの研究が試

みられている。代表的な結果として、平面グラフに対する多項式時間完全マッチング計数アルゴリズム [16] が挙げられる。この結果は、制限された種数を持つグラフ [10, 23], 制限された木幅を持つグラフ [1] などに一般化されている。また、コーダルグラフとそのサブクラスについての計算困難性の考察が Okamoto ら [20] によって行われている。

完全マッチング計数問題に対する近似計算は、大きく分けて2つのアプローチが考えられている。一つ目はマルコフ連鎖モンテカルロ法 (MCMC 法) を利用するアプローチであり、Jerrum らによる同問題への多項式時間乱択近似スキーム (FPRAS) が代表的なアルゴリズムとして知られている [13, 14]。この結果は Bezákova らにより計算時間が改善されている [3]。二つ目のアプローチは行列式の計算への帰着に基づく手法である [11, 15, 6]。この路線における現在最良の結果は Chien らによる計算時間 $O(1.2^n)$ の近似アルゴリズムであり [6], 多項式時間乱択近似スキームが構成可能であるかどうかは現時点では未解決の問題である。

完全マッチング計数に対する $\tilde{O}(2^{o(n)})$ 時間アルゴリズムの存在性についての研究は、Dell らによる結果 [8] が知られている。この結果では SAT に対する指数計算量仮説 (ETH) [12] を仮定した場合², 1 の個数が m 個であるような疎行列の 0-1 パーマネント計算が $\Omega(\exp(m/\log m))$ の時間下限を持つことが示されている。

3 符号理論からの準備

2元有限体 \mathbb{F}_2 上の $n \times m$ 行列 M によって生成される線形符号 C とは、以下の集合によって定められる n 次元ベクトルの集合である。

$$C = \{Mv \mid v \in \mathbb{F}_2^m\}$$

このとき、行列 M は符号 C の生成行列と呼ばれる。言い換えると、線形符号 C は、生成行列 M の各行

²実際には、ETH をより弱めた仮説である、SAT 解の個数の計数が指数時間かかるという仮説 (#ETH) を仮定している。

ベクトルにより張られる線形部分空間と見なすことができる。なお、一般には各行が一次独立でないような生成行列 M も考えることができ、そのような場合 C の次元は n より小さくなる（すなわち、符号語の個数が減少する）。線形符号 C の符号語の個数を $|C|$ で表す。 C が M の行ベクトルにより張られる線形部分空間をであることから、 $|C| = 2^{\text{rank}(M)}$ となる。 C の次元が d であるような符号語長 m の線形符号を特に (m, d) -線形符号と呼ぶ。

線形符号 C に含まれる任意の符号語 $w \in C$ について $w^T H = 0$ を満たす $m \times (m - \text{rank}(M))$ 行列 H を、符号 H の検査行列と呼ぶ。生成行列 M と検査行列 H の間には双対関係があり、検査行列 H を生成行列として定めた符号 C^\perp において、任意の $v \in C^\perp$ について定義より $v^T M = 0$ である。すなわち、 M が C^\perp の検査行列となる。このように定義される符号 C^\perp は元の符号 C の双対符号と呼ばれる。なお、 v を符号 C の任意の符号語、 v^\perp を C^\perp の任意の符号語とすると、定義より明らかに $v^T v^\perp = 0$ が成立する。すなわち主符号の符号語と双対符号の符号語は直交しており、このことから双対符号は主符号の直交補空間をなす事がわかる。

ある符号語における 1 の個数を、その符号語の重みとよぶ。以降、符号 C における重み k の語の個数を $A_C(k)$ で表す。各重みを持つ符号語の個数を並べて得られるベクトル $\{A_C(k)\}$ を符号 C の重み分布と呼ぶ。また、重み分布 $A_C(0), A_C(1), \dots, A_C(m)$ に対する母関数 $F_C(x) = \sum_{w=0}^m A_C(w)x^w$ を、符号 C の重み分布多項式と呼ぶ。

主符号 C と双対符号 C^\perp の重み分布多項式の間には、以下のような関係が成立することが知られている。

定理 1 (マックウィリアムスの恒等式) 語長 m の線形符号 C に対する重み分布多項式 $F_C(x)$ 、およびその双対符号 C^\perp の重み分布多項式 $F_{C^\perp}(x)$ に対して、以下の関係が成立する。

$$F_C(x) = \frac{1}{|C^\perp|} (1+x)^m F_{C^\perp}\left(\frac{1-x}{1+x}\right)$$

この恒等式について係数比較をすることで、主符号と双対符号の重みについての以下の関係式を得ることができる。

$$A_C(i) = \frac{1}{|C^\perp|} \sum_{j=0}^m K_m(j, i) A_{C^\perp}(j)$$

ここで $K_m(j, i)$ は Krawtchouk 多項式と呼ばれる式であり

$$K_m(j, i) = \sum_{k=0}^m (-1)^k \binom{i}{k} \binom{m-i}{j-k}$$

と定義される。

4 カット空間および閉路空間

本節では行列、ベクトルの要素および計算はすべて \mathbb{F}_2 上の演算であるとする。無向グラフ $G = (V, E)$ に対して、その接続行列 M_G は以下のように定義される。

$$A_G(i, j) = \begin{cases} 1 & \text{if } i \text{ が } j \text{ の端点} \\ 0 & \text{それ以外} \end{cases}$$

今、 G の頂点の部分集合 $S \subset V$ を表す n 次元の 0-1 指標ベクトル \mathbf{v}_S を考える。このとき、よく知られた事実として、 m 次元ベクトル $M_G \cdot \mathbf{v}_S$ は頂点集合 V を S と $V - S$ に分割するカットの辺集合を表す。よって、グラフ G の接続行列を生成行列として得られる線形符号は、 G のカット辺集合に対応する指標ベクトルすべてからなる空間と一致する。以下、この符号に対応する線形部分空間をカット空間とよぶ。

G における単純閉路に対応する m 次元指標ベクトルすべてを行ベクトルとして並べた行列を、タイセット行列と呼び、タイセット行列の行ベクトルの一次結合で表せる線形部分空間のことを閉路空間と呼ぶ。閉路空間は、全頂点の次数が偶数であるような全域部分グラフ (の辺集合を表す指標ベクトル) すべてからなる空間である。カット空間と同様に、閉路空間もタイセット行列により定義される線形符号と見な

すことができる。よって、以降の議論では、グラフのカット空間および閉路空間を線形符号の一種と見なし、特に断り無く符号理論の用語を用いる。また、「カット空間 (あるいは閉路空間) の符号語」という言葉は、それに対応するカット辺の集合 (あるいは対応する全域部分偶グラフ) のことを指す場合がある。

以下はカット空間および閉路空間の間の関係としてよく知られている事実である。

事実 1 カット空間および閉路空間は互いに相手の直交補空間である。

この事実は、任意のカットは閉路を構成する辺を偶数回横切る (すなわち、偶数個の共通辺を持つ) ことから直感的に理解できる。符号理論的な観点から見た場合、この事実はカット空間に対応する線形符号および閉路空間に対応する線形符号の間に主双対関係があることを意味する。以降、 $C(G)$ を G の閉路空間に対応する線形符号とし、 $C^\perp(G)$ を同カット空間に対応する線形符号とする。

5 カット重み分布を利用した完全マッチングの数え上げ

5.1 一般のグラフの場合

連結グラフ $G = (V, E)$ の完全マッチングの数え上げを考える。以降の議論では $|V| = n$, $|E| = m$ とし、また n は偶数であるとする (奇数の場合は明らかに完全マッチングは存在しない)。頂点 $v \in V$ の次数を $d(v)$ で表すとする。このとき、以下の定理が成立する。

補題 1 G を奇グラフとする。このとき、 G の完全マッチングの個数は閉路空間 $C(G)$ における重み $m - n/2$ の符号語の個数に等しい。

証明 G の各頂点 $v_0, v_1, v_2, \dots, v_{n-1}$ に対する次数系列が $d(v_0) - 1, d(v_1) - 1, \dots, d(v_{n-1}) - 1$ であるような G の全域部分グラフ $G' = (V, E')$ を考える。

このとき、辺の補集合により誘導される部分グラフ $(V, E \setminus E')$ における全頂点の次数は1, すなわち完全マッチングとなる. また, G 上の任意の完全マッチングに対して, その辺の補集合は明らかに次数系列 $d(v_0)-1, d(v_1)-1, \dots, d(v_{n-1})-1$ を持つ. すなわち, そのような次数系列を持つ全域部分グラフと完全マッチングは一対一対応づけられる. よって, 証明は, 任意の重み $m-n/2$ の符号語 v について, それに対応するグラフ $G(v)$ の頂点次数系列が $d(v_0)-1, d(v_1)-1, \dots, d(v_{n-1})-1$ となることを示せば十分である.

証明は背理法による. v を重み $m-n/2$ の符号語とし, $G(v)$ の次数系列を $d'_0, d'_1, \dots, d'_{n-1}$ とする. 今, ある頂点 v_k についてその次数 d'_k が $d(v_k)-1$ に一致しないと仮定する. 閉路空間の定義より, d'_k は偶数である. また, G が奇グラフであることより, $d(v_k)$ は奇数である. よって, $d'_k \neq d(v_k)-1$ ならば明らかに $d'_k < d(v_k)-1$ である. このとき, $\sum_{i=0}^{n-1} (d'_i - 1) = m - n/2 = \sum_{i=0}^{n-1} (d(v_i) - 1)$ なので, 明らかに $G(v)$ 中に別のある頂点 v_j が存在し, その次数は $d(v_j)-1$ より大きい. d'_j は偶数であるため $d'_j > d(v_j)$ となるが, これは明らかに矛盾である. 以上より示された. \square

補題1は, G が奇グラフの場合にのみ成立する命題であるが, 一般のグラフに G については, 同定理を適用可能な形にグラフを変形させることができる. $G = (V, E)$ を任意の連結なグラフとし, V 中における偶数次数の頂点の集合を V^{even} とする. いま, 以下のようにして得られるグラフ $\hat{G} = (\hat{V}, \hat{E})$ を考える.

- 頂点集合 V に新たな頂点 \hat{v}_1, \hat{v}_2 を追加する. すなわち $\hat{V} = V \cup \{\hat{v}_1, \hat{v}_2\}$.
- 新たに追加した2頂点 \hat{v}_1, \hat{v}_2 間に辺を引き, また, \hat{v}_1 と G 中のすべての偶数次数の頂点に辺を引く, すなわち, $\hat{E} = E \cup \{(v, \hat{v}_1) | v \in V^{\text{even}}\} \cup \{(\hat{v}_1, \hat{v}_2)\}$.

上記の変形の例を図1に示す. この変形により得られるグラフ \hat{G} に対して, 以下の補題が成立する.

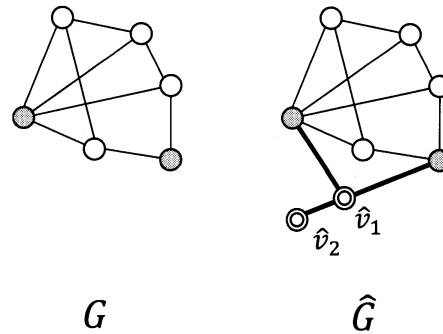


図 1: \hat{G} の構成例

補題2 任意の連結なグラフ G に対して, \hat{G} は奇グラフであり, また \hat{G} と G の完全マッチングの個数は等しい.

証明 まず, \hat{G} が奇グラフであることを示す. \hat{v}_2 の頂点は1であり, また, G 中のすべての偶数次数の頂点は \hat{v}_1 への辺が追加されることにより奇数次数となるため, 証明は \hat{v}_1 の次数が奇数であること, すなわち, G 中における偶数頂点の個数が偶数であることを示せば十分である. 握手補題より全頂点の次数の総和は必ず偶数となるので, G における奇数次数の頂点は偶数個存在する. G の頂点数は偶数であるので, G における偶数次数の頂点数は偶数個存在する. 以上より示された.

次に G と \hat{G} の完全マッチングの個数が一致することを示す. \hat{G} において追加された頂点は \hat{v}_1 および \hat{v}_2 であり, このうち \hat{v}_2 は次数1である. よって, \hat{G} の完全マッチングにおいて, \hat{v}_2 に接続するマッチング辺は (\hat{v}_1, \hat{v}_2) の一通りのみであり, \hat{G} の完全マッチングから辺 (\hat{v}_1, \hat{v}_2) を取り除いたグラフは必ず G の完全マッチングとなる. このことより, \hat{G} の完全マッチングと G の完全マッチングの間に一対一対応の関係を定義することが可能であり, 結果として G と \hat{G} の完全マッチングの個数は一致する. \square

上記補題およびマックウィリアムスの恒等式より, 以下の定理を得ることができる.

定理 2 任意のグラフ G に対して, \hat{G} のカット重み分布 $A_{C^\perp(\hat{G})}(0), A_{C^\perp(\hat{G})}(1), \dots, A_{C^\perp(\hat{G})}(m)$ を $O(f(n))$ 時間で計算するアルゴリズムが存在するとき, G の完全マッチングの個数は $O(f(n) + \text{poly}(n))$ 時間で計算可能である.

証明 n 頂点の連結なグラフの接続行列はランク $n-1$ であることが知られているため, $|C(\hat{G})| = 2^{n-1}$ である. また, マックウィリアムス恒等式に現れるすべての値は $O(n)$ ビットで表現可能であることは容易に確かめることができ, その四則演算も n の多項式時間で実行可能である. よって, マックウィリアムス恒等式を利用した $A_{C(\hat{G})}(n)$ の計算は n の多項式時間で可能であり, 定理が成立する. \square

5.2 二部グラフの場合

前節に述べた帰着では, G が二部グラフの場合, \hat{G} が一般には二部グラフとならないという問題点がある. この問題を解決するため, 本節では任意の二部グラフ G からの, 完全マッチングの個数を保存する二部奇グラフ \tilde{G} の構成を与える. この構成を用いることで, 二部グラフの完全マッチングの数え上げを二部グラフのカット重み分布の計算に帰着することが可能である.

$G = (V_1, V_2, E)$ を二部グラフとして, V_1 および V_2 における偶数次数の頂点集合をそれぞれ $V_1^{\text{even}}, V_2^{\text{even}}$ とする. 今は, 完全マッチングを持つような G に興味があるので, $|V_1| = |V_2|$ であることを仮定してよい. ここで, 以下の補題が成り立つ.

補題 3 $|V_1^{\text{even}}|$ と $|V_2^{\text{even}}|$ の偶奇は一致する.

証明 $|E|$ が奇数のとき, 任意の $i \in \{1, 2\}$ について, $\sum_{v \in V_i^{\text{even}}} d(v)$ はつねに偶数なので, $\sum_{v \in V \setminus V_i^{\text{even}}} d(v)$ は奇数である. $V \setminus V_i^{\text{even}}$ に含まれる頂点の次数は奇数なので, 結果として $|V \setminus V_i^{\text{even}}|$ が奇数であることを意味する. $|E|$ が偶数の場合も同様に示すことができる. \square

以下, G に対応するグラフ $\tilde{G} = (\tilde{V}_1, \tilde{V}_2, \tilde{E})$ の構成を示す. 構成は, 補題 3 より, $|V_1^{\text{even}}| (= |V_2^{\text{even}}|)$ の偶奇によって場合分けされる.

$|V_1^{\text{even}}|$ および $|V_2^{\text{even}}|$ が偶数の場合

- $|V_i^{\text{even}}| \geq 2$ ならば, 頂点集合 $V_{1-i}, V_i (i \in \{0, 1\})$ それぞれに対して, 新たな頂点 $\tilde{v}_{1-i,1}, \tilde{v}_{i,2}$ を追加する.
- $i \in \{0, 1\}$ に対して, $\tilde{v}_{1-i,1}$ と $\tilde{v}_{i,2}$ を辺で接続する. また, $\tilde{v}_{1-i,1}$ と V_i^{even} 中のすべての頂点を接続する.

$|V_1^{\text{even}}|$ および $|V_2^{\text{even}}|$ が奇数の場合

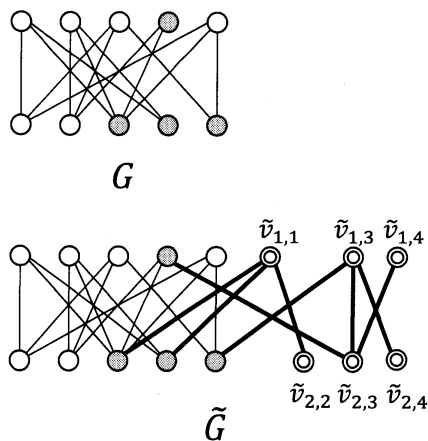
- $|V_i^{\text{even}}| \geq 1$ ならば, 頂点集合 $V_{1-i}, V_i (i \in \{0, 1\})$ それぞれに対して, 新たな頂点 $\tilde{v}_{i,1}, \tilde{v}_{1-i,2}$ を追加する. さらに, $|V_i^{\text{even}}| \geq 3$ ならば, $\tilde{v}_{1-i,3}, \tilde{v}_{i,4}$ を追加する.
- $i \in \{0, 1\}$ に対して, $\tilde{v}_{i,1}$ と $\tilde{v}_{1-i,2}, \tilde{v}_{i,3}$ と $\tilde{v}_{1-i,4}$ を辺で接続する. また, $\tilde{v}_{1,3}$ と $\tilde{v}_{2,3}$ を接続する. さらに V_i^{even} から任意の $|V_i^{\text{even}}| - 1$ 個の頂点を選び, $\tilde{v}_{1-i,2}$ と接続する. $\tilde{v}_{1-i,2}$ と接続しなかった頂点は, $\tilde{v}_{1-i,3}$ と接続する.

上記の構成の例を図 2 に示す. グラフ \tilde{G} に対して, 以下の定理を得ることができる.

補題 4 任意の連結なグラフ G に対して, \tilde{G} は奇グラフであり, また \tilde{G} と G の完全マッチングの個数は等しい.

定理 3 任意のグラフ G に対して, \tilde{G} のカット重み分布 $A_{C^\perp(\tilde{G})}(0), A_{C^\perp(\tilde{G})}(1), \dots, A_{C^\perp(\tilde{G})}(m)$ を $O(f(n))$ 時間で計算するアルゴリズムが存在するとき, G の完全マッチングの個数は $O(f(n) + \text{poly}(n))$ 時間で計算可能である.

証明は補題 2, および定理 2 とほぼ同様である.

図 2: \tilde{G} の構成例

6 3-正則二部グラフに対するカット重み分布の計算

本節では、3-正則二部グラフに対するカット重み分布の計算を行う $\tilde{O}(2^{5n/12})$ 時間のアルゴリズムについて述べる。以降ではまず、 $\tilde{O}(2^{n/2})$ 時間で重み分布を計算するアルゴリズムを示したのち、それを改良して所望のアルゴリズムを得る。

6.1 $\tilde{O}(2^{n/2})$ 時間アルゴリズム

本節では、連結な 3-正則二部グラフに対して $\tilde{O}(2^{n/2})$ 時間でカット重み分布を計算するアルゴリズムを提案する。

入力として与えられる 3-正則連結二部グラフを $G = (V_1, V_2, E)$ とする。3-正則であることより、 $|E| = 3n/2$ であり、 $|V_1| = |V_2|$ である。カットは $V = V_1 \cup V_2$ の部分集合 S で表せる (S と $V \setminus S$ の間の辺がカット辺となる) ので、以降はあるカットの選択を頂点の (proper でない) 2-彩色として考える。 S に含まれる頂点は値 1 に彩色、 $V \setminus S$ に含まれる頂点の彩色は値 0 に彩色されていると見なす。なお、グラフが連結であることから、彩色によるカットの表現を用いた場合、あるカットセットに対する彩色

は正確に 2 通り (S と $V \setminus S$ を入れ替えても同じカットセットを表す) 存在するため、彩色のパタン数の半分が実際のカットの個数に対応する。なお、以降の議論では、しばしばグラフの一部分のみを彩色することを考える。頂点の部分集合 $V' \subseteq V$ に対してのみ 1 あるいは 0 の色を割り当てることを、 V' への (部分的な) 彩色と呼ぶこととする。また、頂点 v の隣接頂点の集合を $N(v)$ で表す。

今、重み k のカットを計数したいとする。提案するアルゴリズムの背景となるアイデアは、 V_1 への部分的な彩色を固定した下で、カットの重みが k となるようなグラフ全体への彩色 (すなわち、残りの頂点 V_2 への部分的な彩色) のパタン数を n の多項式時間で計数可能であるという事実である。以下、そのアイデアの詳細を述べる。

今、ある彩色 X が V_1 に与えられているとする。 G は 3-正則二部グラフなので、 V_2 に含まれる各頂点 v に対して、 $N(v)$ 中の頂点はすべて V_1 に属しており、彩色済みである。そのため、 v に接続している 3 本の辺がカットに属するかどうかは、 v の彩色によってのみ決定される。ここで、 $N(v)$ がどのように彩色されているかについては、以下の 2 つの場合が存在する。

1. $N(v)$ 中の頂点がすべてが同じ値で彩色されている。
2. $N(v)$ 中の頂点のうち 2 点が同じ値で、残り一点が異なる値で彩色されている。

以降、前者に当てはまるような v を 3-頂点と呼び、後者に当てはまるような v を 2-頂点と呼ぶ。直感的には、ある頂点 v が 3-頂点であるということは、 v を的確に彩色することで隣接辺のうち 3 本または 0 本をカット辺に含めることが可能であるということであり、2-頂点であるということは、接続辺のうち 2 本または 1 本をカット辺に含めることが可能であるということである。図 3 に V_1 への部分的な彩色の例を示す。この例において、2-頂点数および 3-頂点数の数はそれぞれ 2, 4 である。 V_1 への 3-頂点 (2-頂点) v について、その隣接辺のうち 3 辺 (2 辺) がカッ

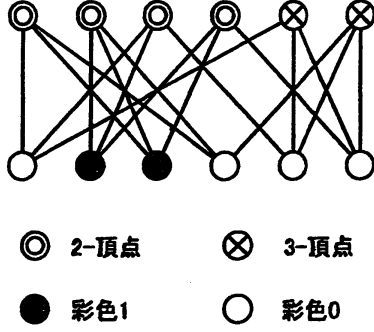


図 3: V_1 への部分的な彩色, および 2-頂点と 3-頂点の例

ト辺に含まれるように v を彩色するとき, v は多数派に彩色されていると呼ぶ.

V_1 への部分的な彩色 X が与えられたもとでの 2-頂点の個数が z であるとする. $|V_1| = n/2$ なので, 3-頂点の個数は $n/2 - z$ であり, このとき,

$$\begin{cases} 3j_3 + 2j_2 + (z - j_2) = k \\ 0 \leq j_3 \leq n/2 - z \\ 0 \leq j_2 \leq z \end{cases} \quad (1)$$

を満たす整数 j_3 および j_2 が存在するならば, 3-頂点のうちの任意の j_3 個, および 2-頂点のうちの任意の j_2 を多数派に彩色することで, サイズ k のカットを得ることができる. 条件 1 を満たす j_3, j_2 の組の集合を $P(k, z)$ とする (すなわち, $P(k, z) = \{(j_3, j_2) | 3j_3 + j_2 = k - z, 0 \leq j_3 \leq n/2 - z, 0 \leq j_2 \leq z\}$) と, V_1 への彩色 X のもとで, カット重みが k となるようなグラフ全体の彩色の個数は

$$\sum_{(j_3, j_2) \in P(k, z)} \binom{n/2 - z}{j_3} \binom{z}{j_2}$$

に正確に一致する. ここで, 上記の式は z および k の関数なので, その値は V_1 への彩色によって決まる 2-頂点数 z のみに依存して決定される. すなわち, V_1 への異なる 2 つの彩色 X, Y に対して, その 2-頂点数が等しければ, 上式の値は X, Y ともに等しくなる. よって, 以下の補題を得ることができる.

補題 5 $G = (V_1, V_2, E)$ を 3-正則 2 部グラフとする. 2-頂点の個数が z であるような V_1 への彩色のパターン数を $Q(z)$ とすると, 重み k のカットの個数 $A_{C^\perp(G)}(k)$ について以下の式が成立する.

$$A_{C^\perp(G)}(k) = \frac{1}{2} \sum_{z=0}^{n/2} \left(Q(z) \sum_{(j_3, j_2) \in P(k, z)} \binom{n/2 - z}{j_3} \binom{z}{j_2} \right)$$

この補題から, カット重み分布を求める以下のようなアルゴリズムを得ることができる.

1. V_1 のすべての彩色を列挙し, $Q(0), Q(1), \dots, Q(n/2)$ を計算する.
2. 補題 5 の式に従って $A_{C(G)}(k)$ をすべての k ($0 \leq k \leq m$) について計算する.

補題 5 の式に現れる数値はいずれも $2^{O(n)}$ の上界を持つので, 任意の四則演算は n の多項式時間で計算が可能である. よって, $Q(z)$ の分布が与えられたもとでの上式の計算は n の多項式時間で可能である. また, V_1 の頂点数は $n/2$ であるので, すべての彩色パターンの個数は $2^{n/2}$ である. 1 つのパターンの列挙およびそのパターンにおける 2-頂点の個数の計算は n の多項式時間で計算可能である. また, 列挙したパターンを記憶しておく必要はないので, この過程において使用されるメモリ量は n の多項式サイズである. よって, このアルゴリズムは $\tilde{O}(2^{n/2})$ 時間, 多項式メモリスペースで, サイズ k のカットの重み分布を計算する.

定理 4 $G = (V, E)$ を 3-正則グラフとする. G のカット重み分布 $A_{C^\perp(G)}(0), A_{C^\perp(G)}(1), \dots, A_{C^\perp(G)}(m)$ は $\tilde{O}(2^{n/2})$ 時間, $\text{poly}(n)$ ビットのメモリ空間で計算可能である.

6.2 分布 $Q(z)$ の高速な計算

前節において提案したアルゴリズムのボトルネックは, V_1 の彩色の全列挙による分布 $Q(z)$ の計算で

ある。本節では、列挙の個数を削減することで、同部分の高速化を行う。本節の高速化をここまでの議論に組み合わせることで、最終的に $\tilde{O}(2^{5n/12})$ 時間の完全マッチング計数アルゴリズムを得ることができる。

頂点集合 $S \subseteq V_1$ を、集合中のどの2頂点も隣接頂点を共有しない、すなわち $\forall u, v \in S : N(u) \cap N(v) = \emptyset$ であるような V_1 の部分集合と定める。また、 $V_1 \setminus S = \hat{V}_1$ と定める。定義より明らかに、 V_2 の各頂点は高々1つの S に属する隣接頂点を持つ。

$Q(z)$ の計算を高速化するための基本的なアイデアは前節とほぼ同様である。核となる事実、 \hat{V}_1 への彩色を固定したもとの、2-頂点数が z となるような V_1 への彩色パターン数 (すなわち S への彩色パターン数) を多項式時間で計算できるということである。今、ある部分的な彩色 X を \hat{V}_1 に与えたと仮定する。このとき、 V_2 の任意の頂点 v は、 $N(v)$ の彩色状況により以下のいずれかの場合に分類される。

1. $N(v)$ はすべて彩色済みであり、2-頂点である。
2. $N(v)$ はすべて彩色済みであり、3-頂点である。
3. $N(v)$ のうち1頂点が S に属しており、残りの2点は異なる色で彩色されている。
4. $N(v)$ のうち1頂点が S に属しており、残りの2点はいずれも色1で彩色されている。
5. $N(v)$ のうち1頂点が S に属しており、残りの2点はいずれも色0で彩色されている。

以降、ケース1または3に属するような頂点を(2,2)-頂点と呼ぶ。また、ケース2, 4, 5に属する頂点をそれぞれ(3,3)-頂点、(3,2)-頂点、(2,3)-頂点と呼ぶ。 (a,b) -頂点は、 S に含まれる隣接頂点が存在するならば、それを1に彩色することで a -頂点、0に彩色することで b -頂点になるということを意味している。また、 S に含まれる隣接頂点を持たない場合は、それは(2,2)-頂点か(3,3)-頂点のいずれかとなる。

今、 \hat{V}_1 への彩色 X が与えられたとき、最終的に2-頂点数の個数が z となるような S の彩色を求めるこ

とを考える。 X に対する、(2,2)-頂点および(3,3)-頂点の集合をそれぞれ $T_2(X), T_3(X)$ とする。これらの頂点は S の彩色内容によらず、それぞれ2-頂点、3-頂点となる。よって、以降は頂点集合 $T(X) = V_2 \setminus (T_2(X) \cup T_3(X))$ が S の彩色により2-頂点、3-頂点のいずれになるかのみ注目する。我々の目的は2-頂点数が z となるように S への部分的な彩色を調整することであるので、 $T(X)$ 中の集合のうち、 $z - |T_2(X)|$ 個を2-頂点にするような S の彩色方法を考える必要がある。これは、 $T(X)$ 中の頂点のうち、2-頂点となるものの個数が3-頂点となるものの個数より

$$\begin{aligned} (z - |T_2(X)|) - (|T(X)| - (z - |T_2(X)|)) \\ = 2z - |T(X)| - 2|T_2(X)| \end{aligned} \quad (2)$$

個だけ多くなるような彩色を考えるということと等価である。

S 中の任意の頂点 v について、 $N(v)$ 中に含まれる(2,3)-頂点および(3,2)-頂点の個数をそれぞれ $\delta_2(v)$ および $\delta_3(v)$ とし、 $|\delta_2(v) - \delta_3(v)| = i$ であるような頂点 $v \in S$ の集合を $S_i(X)$ とする。ある頂点 $v \in S_i(X)$ について、 $N(v)$ における2-頂点の個数が多くなるように v を彩色するとき、 v を多数派で彩色するという。例えば、 v の3隣接頂点がそれぞれ(2,3)-頂点、(3,2)-頂点、(3,2)-頂点のとき、彩色0が v への多数派の彩色となる。

\hat{V}_1 に彩色 X が与えられているもとの、 S に対するある彩色 Y を与えられたとする。このとき、 $T(X)$ において生じる2-頂点、3-頂点の個数をそれぞれ $z_2(X, Y), z_3(X, Y)$ とする。今、集合 $S_i(X)$ のうち j_i 個の頂点が多数派で彩色されているとすると、 $S_i(X)$ の定義より、

$$\begin{aligned} z_2(X, Y) - z_3(X, Y) \\ = j_1 + 2j_2 + 3j_3 - (|S_1(X)| - j_1) \\ - 2(|S_2(X)| - j_2) - 3(|S_3(X)| - j_3) \\ = \sum_{i=1}^3 (2i \cdot j_i + i|S_i(X)|) \end{aligned} \quad (3)$$

が成立する。よって、式2および3より、 Y がグラ

全体として 2-頂点を z 個持つ彩色となるための必要十分条件は,

$$\begin{aligned}
 z_2(X, Y) - z_3(X, Y) \\
 &= 2z - |T(X)| - 2|T_2(X)| \\
 &\Leftrightarrow \sum_{i=1}^3 (2i \cdot j_i + i|S_i(X)|) \\
 &= 2z - |T(X)| - 2|T_2(X)| \quad (4)
 \end{aligned}$$

である。言い換えるならば, 上記条件を満たすような j_1, j_2, j_3 ($0 \leq j_i \leq |S_i(X)|$) に対して, $S_i(X)$ ($i \in \{1, 2, 3\}$) に含まれる頂点のうち任意の j_i 個を多数派で彩色することで, 2-頂点数が z であるような V_1 の部分的な彩色を得ることができる。このことから, \hat{V}_1 へと彩色 X を与えたもとの, 2-頂点数が z 個となるような V_1 への部分的な彩色のパターン数は

$$\sum_{(j_1, j_2, j_3) \in P'(X, z)} \binom{|S_1(X)|}{j_1} \binom{|S_2(X)|}{j_2} \binom{|S_3(X)|}{j_3}$$

と書ける。ここで $P'(X, z)$ は式 4 および $0 \leq j_i \leq |S_i(X)|$ ($i \in \{1, 2, 3\}$) を満たすような整数の 3 項組 (j_1, j_2, j_3) すべてからなる集合と定義している。

$P'(X, z)$ のサイズは高々 $O(n^3)$ であるので, 上記の式は明らかに n の多項式時間で計算が可能である。よって, \hat{V}_1 への可能なすべての彩色 X について上記の式を計算することで, $Q(z)$ の分布を得ることができる。

6.3 集合 S の選択

前述のアルゴリズムでは, \hat{V}_1 に対する部分的な彩色を全列挙することで $Q(z)$ の分布を得ることができるので, その計算時間は $\tilde{O}(2^{|\hat{V}_1|}) = \tilde{O}(2^{n/2 - |S|})$ である。そのため, できるだけサイズの大きい集合 S を選択して, アルゴリズムを実行することが望ましい。以下に示す単純な手続きを利用することで, 任意の 3-正則二部グラフに対して, 少なくとも $|S| \geq n/12 - 1$ となるような V_1 の部分集合 S を構成することが可能である。

1. V_1 中の任意の 1 頂点 v を S に加える。
2. S に含まれる任意の頂点との距離が 4 以上, かつ少なくともある 1 頂点との距離が正確に 4 であるような $V_1 \setminus S$ 内の頂点を選び, S に加える。
3. 加えられる頂点が無くなるまでステップ 2 を繰り返す。

補題 6 上記アルゴリズムにより選択された集合 S について, S 中の任意の 2 頂点は共通の隣接頂点を持たない。また, $|S| \geq n/12 - 1$ である。

証明 S 中の任意の 2 頂点が共通の隣接頂点を持たないことは, それらの距離が 4 以上離れていることから明らかである。以降, $|S| \geq n/12 - 1$ を示す。ある頂点 $u \in V_1$ に対して, u からの距離が 2 であるような頂点の集合を $N_2(u)$ とする。 $N_2(u)$ は, u を S に追加することでそれ以降 S に加えることができなくなる頂点の集合である。手続きの実行において追加される順番に S の頂点を並べた系列を $u_1, u_2, \dots, u_{|S|}$ とする。手続きの終了時点において S にはもう新たな頂点が追加できないので, 任意の頂点 $v \in V_1$ は, 既に S に含まれているか, ある i に対して $v \in N_2(u_i)$ であるかのいずれかである。よって,

$$|S| + |\cup_{i=0}^{|S|} N_2(u_i)| = n/2 \quad (5)$$

が成立する。 $i = 0$ を除いて, u_i が S に追加されるとき, ある u_l ($0 \leq l \leq i$) に対して u_i と u_l の距離が正確に 4 であることが成り立つので, u_i と u_l の間の距離 4 のパス上にある V_1 中の頂点は, $N_2(u_i)$ および $N_2(u_l)$ のいずれにも含まれる。すなわち, 任意の $i > 0$ についてある l が存在し, $|N_2(u_i) \cap N_2(u_l)| \geq 1$ が成立する。よって, 集合 $N_2(u_0), N_2(u_1), \dots, N_2(u_{|S|})$ は, 少なくとも $|S| - 1$ 個の要素を重複して含んでいることがわかり, このことから, 不等式

$$|\cup_{i=0}^{|S|} N_2(u_i)| \leq \sum_{i=0}^{|S|} |N_2(u_i)| - |S| + 1$$

を得る。グラフが 3-正則二部グラフであることから, 任意の u_i について, $|N_2(u_i)| = 6$ であるので, この

不等式は

$$|\cup_{i=0}^{|S|} N_2(u_i)| \leq 5|S| + 1$$

と変形でき、式5と組み合わせることで $n/2 \leq 6|S| + 1$ が成立する。これより $|S| \geq n/12 - 1$ を得る。□

上記アルゴリズムで得られる集合 S を利用することで、以下の定理を得る。

定理 5 $Q(z)$ の分布は、 $\tilde{O}(2^{5n/12})$ 時間、 $\text{poly}(n)$ ビットメモリ空間で計算可能である。

定理 2, 定理 5, および補題 5 を組み合わせることで、以下の系が導かれる。

系 1 3-正則二部グラフの完全マッチングの個数は、 $\tilde{O}(2^{5n/12})$ 時間、 $\text{poly}(n)$ ビットメモリ空間で計算可能である。

7 むすび

本論文では、完全マッチング計数問題に対する、符号理論からのアプローチに基づく新たなアルゴリズムを提案した。提案アルゴリズムは、3-正則二部グラフに対して、多項式メモリスペース、計算時間 $\tilde{O}(2^{5n/12})$ を達成している。この結果は多項式メモリスペースのアルゴリズムとしては我々の知る限り最速である。提案手法では線形符号の双対性を利用して完全マッチングの数え上げをカットの重み分布計算に帰着することで高速なアルゴリズムを得ている。この手法は数え上げ問題に対する新しいアプローチとして興味深く、同様の手法の他の数え上げ問題への適用、および一般のマッチング計数への応用等、さらなる発展が考えられる。

参考文献

- [1] S. Arnborg, J. Lagergren, and D.f Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12:308–340, 1991.
- [2] E. T. Bax and J. Franklin. A permanent algorithm with $\exp[\omega(n^{1/3}/2\ln(n))]$ expected speedup for 0-1 matrices. *Algorithmica*, 32(1):157–162, 2002.
- [3] I. Bezáková, D. Štefankovič, V. V. Vazirani, and E. Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. In *Proc. of the 17th annual ACM-SIAM symposium on Discrete algorithm (SODA)*, pages 900–907, 2006.
- [4] A. Björklund. Counting perfect matchings as fast as ryser. In *Proc of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2012. to appear.
- [5] A. Björklund and T. Husfeldt. Exact algorithms for exact satisfiability and number of perfect matchings. *Algorithmica*, 52:226–249, 2008.
- [6] S. Chien, L. Rasmussen, and A. Sinclair. Clifford algebras and approximating the permanent. In *Proc. of the 34th annual ACM symposium on Theory of computing (STOC)*, pages 222–231, 2002.
- [7] P. Dagum and M. Luby. Approximating the permanent of graphs with large factors. *Theoretical Computer Science*, 102:283–305, 1992.
- [8] H. Dell, T. Husfeldt, and M. Wahlén. Exponential time complexity of the permanent and the tutte polynomial. In *Proc. of the 37th international colloquium conference on Automata, languages and programming*, volume LNCS 6198, pages 426–437, 2010.
- [9] Fedor V. Fomin and Kjartan Høie. Pathwidth of cubic graphs and exact algorithms. *Information Processing Letters*, 97:191–196, March 2006.

- [10] A. Galluccio and M. Loeb. On the theory of pfaffian orientations. i. perfect matchings and permanents. *Electronic Journal of Combinatorics*, 6, 1999.
- [11] C. D. Godsil and I. Gutman. On the matching polynomial of a graph. In *Algebraic Methods in Graph Theory*, pages 241–249. 1981.
- [12] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
- [13] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18:1149–1178, 1989.
- [14] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51:671–697, 2004.
- [15] N. Karmarkar, R. Karp, R. Lipton, L. Lovász, and M. Luby. A monte-carlo algorithm for estimating the permanent. *SIAM Journal on Computing*, 22:284–293, 1993.
- [16] P. Kasteleyn. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.
- [17] M. Koivisto. Partitioning into sets of bounded cardinality. In *Proc. of 4th International Workshop on Parameterized and Exact Computation (IWPEC)*, volume LNCS 5917, pages 258–263, 2009.
- [18] J. Nederlof. Inclusion exclusion for hard problems, 2008.
- [19] J. Nederlof. Fast polynomial-space algorithms using mobius inversion: Improving on steiner tree and related problems. available at <http://www.uib.no/People/jne061/Steinerfull.pdf>, 2010.
- [20] Y. Okamoto, R. Uehara, and T. Uno. Counting the number of matchings in chordal and chordal bipartite graph classes. In *Proc. of 35th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 296–307, 2009.
- [21] H.J. Ryser. *Combinatorial Mathematics*. The Carus mathematical monographs. Mathematical Association of America, 1963.
- [22] R. A. Servedio and A. Wan. Computing sparse permanents faster. *Information Processing Letters*, 96:89–92, November 2005.
- [23] G. Tesler. Matchings in graphs on non-orientable surfaces. *Journal of Combinatorial Theory Series B*, 78:198–231, 2000.
- [24] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.